



Development of a yoga posture coaching system using an interactive display based on transfer learning

Chhaihuoy Long¹ · Eunhye Jo² · Yunyoung Nam³

Accepted: 6 September 2021 / Published online: 20 September 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Yoga is a form of exercise that is beneficial for health, focusing on physical, mental, and spiritual connections. However, practicing yoga and adopting incorrect postures can cause health problems, such as muscle sprains and pain. In this study, we propose the development of a yoga posture coaching system using an interactive display, based on a transfer learning technique. The 14 different yoga postures were collected from an RGB camera, and eight participants were required to perform each yoga posture 10 times. Data augmentation was applied to oversample and prevent over-fitting of the training datasets. Six transfer learning models (TL-VGG16-DA, TL-VGG19-DA, TL-MobileNet-DA, TL-MobileNetV2-DA, TL-InceptionV3-DA, and TL-DenseNet201-DA) were exploited for classification tasks to select the optimal model for the yoga coaching system, based on evaluation metrics. As a result, the TL-MobileNet-DA model was selected as the optimal model, showing an overall accuracy of 98.43%, sensitivity of 98.30%, specificity of 99.88%, and Matthews correlation coefficient of 0.9831. The study presented a yoga posture coaching system that recognized the yoga posture movement of users, in real time, according to the selected yoga posture guidance and can coach them to avoid incorrect postures.

Keywords Yoga · Posture classification · Transfer learning · Self-coaching system · Real-time instruction feedback

✉ Yunyoung Nam
ynam@sch.ac.kr

Chhaihuoy Long
longchhaihuoy@gmail.com

Eunhye Jo
eunhyejo@sch.ac.kr

¹ Department of ICT Convergence, Soonchunhyang University, Asan 31538, South Korea

² ICT Convergence Research Center, Soonchunhyang University, Asan 31538, South Korea

³ Department of Computer Science and Engineering, Soonchunhyang University, Asan 31538, South Korea

1 Introduction

Yoga, which originates from India, is a form of healthy exercise that encourages harmony between the body (physical) and mind (mental). It is one of the oldest sciences in the world that is known to be effective for sustaining and preserving physical and mental health, and spiritual evolution [1]. Yoga practicing provides the benefits of improved flexibility, energy levels, sleep, posture, muscle strength, circulatory and cardio health, athletic performance, and reduction of injuries as well as chronic pain. Yoga is effective in treating psychiatric disorders and is more effective than therapeutic exercise for chronic lower back pain [2, 3]. The survey respondents (89.4%) perceived that yoga aids the relief of joint pain, muscle pain, headaches, and depression [4]. However, improper yoga positions have been shown to cause neck, waist, shoulder, wrist, knee, sprains, and muscle pain. Specifically, a survey of 34 countries in 2007 found that practicing yoga incorrectly generally causes neck injuries [5]. Moreover, of the 46.6% injuries resulting from poor practice, injuries to the lower limbs (21.9%), and sprains (45.03%) were observed from a survey of yoga-related injury cases (2001–2014) [6]. The most common type of injury was found to be a sprain (23/67, 34%) with the lower extremity (27/67, 42%), in one Canadian study (1991–2010) [7].

Several studies have recently been actively conducted to combine artificial intelligence (AI) [8, 9] and yoga training to prevent such injuries. In previous studies, eye-free yoga was studied using Microsoft Kinect as a yoga instructor [10], and computer-assisted vision of a yoga learning system was studied in [11]. AI-based yoga posture estimation was proposed by Girija et al. [12], who focused on a survey study of posture estimation that was applied to an Android application. They presented an Android application graphical user interface (GUI) that took the user video and predicted the posture angle, using the PoseNet algorithm for comparison with the model training database. The resulting feedback given was based on the calculated angle. Moreover, real-time activity was proposed, through assistance being embedded into the yoga mat, by Chinnaiah et al. [13]. This approach can identify pressure nodes through the use of force sensitive resistor (FSR) sensors utilized. The pressure data collected are utilized with an algorithm for pattern detection and biofeedback of posture correction of a subject is provided in real-time.

The reminder of this manuscript is organized as follows. Section 2 is the related work and Sect. 3 is the material and methods of this study. The experiment result is provided in Sect. 4. The whole work is discussed in Sect. 5 and concluded in Sect. 6.

2 Related work

Several companies have developed various products associated with technology in the sports and exercise domain. Wearable X company [14] proposed a wearable product called NADI X-Smart Yoga Pants, which could guide exercise form

via a mobile application. Another company, SmartMat [15], established an intelligent yoga mat with an advanced sensor to detect the pressure node of posture on the mat and provide real-time feedback on how the user has performed the yoga posture, i.e., correctly or incorrectly, via a mobile application. YogaNotch [16] came up with a yoga wearable device assistant, placed on the body, which provides audio feedback on alignments when the user practices yoga at home. One more popular exercise guidance product in this current era is MIRROR [17], which is a movable mirror for use in a home gym that enables exercises to be viewed together with the user's reflection.

However, yoga posture recognition via a mobile application [12] is complicated, with narrow monitoring for users to achieve the correct feedback. In [13], the proposed yoga mat with a pressure sensor was inadequate for correcting whole-body posture. Furthermore, the other products developed in [14–16] are insufficient for exercising with reflection. In addition, the product that came with a reflection monitor, i.e., MIRROR [17], still has issues as it provides a camera without any body-posture detection or correction. Additionally, MIRROR is expensive to use, including product prices and monthly training class subscriptions, which could be a barrier to many.

Yoga classification plays a significant role in many studies using different approaches, such as posture estimation using the OpenPose algorithm [18] and posture recognition by machine learning (ML) and deep learning (DL) techniques. Palanimeera et al. [19] presented yoga posture classification using ML for multiple people to detect real-time posture by using posture estimation with a 3D posture from an RGB camera. In [19], they considered 12 yoga postures, from the sun salutation asana, captured by a webcam camera. The method created a skeleton, applied featured extraction, and classified the sun salutation yoga postures through ML models as well as support vector machine (SVM), k-nearest neighbors (kNN), naïve Bayes, and logistic regression with a resulting of 96%.

Kumar et al. [20] proposed yoga posture estimation by using keypoint detection method, called the OpenPose algorithm, on the public yoga asanas dataset [21]. The method was collected in a video frame of six yoga asanas, including cobra posture, lotus posture or sitting posture, corpse posture, mountain posture, triangle posture, and tree posture.

Nagalakshmi et al. [22] applied the learning of 3D landmark points, from a single image, using skinned multi-person linear (SMPL) and an encoded architecture of classification models. These models included kNN, SVM, and other deep learning models such as AlexNet, VggNet, and ResNet. Each model was utilized with the proposed dataset that was based on a collection of 13 different yoga postures. The dataset includes a half-camel posture, standing half-forward bend posture, butterfly posture, cobra posture, bridge posture, sitting posture, standing forward bend posture, child posture, corpse posture, mountain posture, tree posture, triangle posture, and twisted posture. Classification was evaluated with an accuracy of 83%.

Chen et al. [23] introduced a computer-assisted self-training system, using Kinect, with a user's body contour and star skeleton extraction for asana recognition, which took three yoga postures: tree posture, warrior 3 posture, and downward-facing dog posture. They achieved an overall accuracy of only 82.84% during their

experiments. The previous research provided a self-training system that estimated the yoga posture by extracting the visual features from image data to analyze the 12 yoga postures, including tree posture, warrior 3 posture, warrior 2 posture, warrior 1 posture, downward-facing dog posture, extended hand-to-big-toe posture, chair posture, full boat posture, cobra posture, side plank posture, plank posture, and lord of the dance posture, to correct yoga posture performance [24]. They captured the user body map to apply the topological skeleton to extract the feature points of the practitioner's body and visualized the posture instructions to adjust the correct posture. Posture recognition was evaluated by a yoga expert frame by frame, achieving 93.45% accuracy.

Manisha et al. [25] proposed a new dataset, called Yoga-82, for the fine-grained classification of human yoga postures. They modified the DenseNet201 model to classify yoga postures within their proposed Yoga-82 dataset and achieved the top-1 score of image classification performance at 79% of 82 multi-classes.

The purpose of this study was to develop a yoga posture coaching system based on transfer learning. We collected yoga posture data for 14 yoga posture classifications and proposed a yoga posture coaching system, using an interactive display, based on transfer learning that pre-trained weights were from the CNN architecture model to recognize a yoga posture in real time. Furthermore, our yoga posture coaching system provided posture instruction feedback when a user performed yoga in front of a yoga posture coaching system.

3 Materials and methods

3.1 Data acquisition

There is limited access to the dataset available for yoga recognition, so we collected the yoga posture images from eight volunteers (6 women and 2 men), some of whom had never experienced yoga before performing each of the 14 yoga postures 10 times. The 14 different yoga postures included bridge posture, cat-cow posture, child posture, cobra posture, corpse posture, downward-facing dog posture, sitting posture, extended side angle posture, warrior II posture, and warrior I posture, as shown in Fig. 1. Yoga datasets were captured using an HD 1080p Logitech c920 pro webcam with an image resolution of 720×960. The total amount of collected data was 1,120 images, which were divided into 60 % for the training set, 20% for the validation set, and 20 % for the testing set for yoga posture classification.

3.2 Data augmentation

Data augmentation (DA) is used to boost the model performance and solve overfitting by increasing the quantity of the data training set [26]. According to the large size of the training set required for deep learning model training, DA is used in this study. We applied the DA, including shear augmentation in the range of 0.2, zoom at



Fig. 1 Sample of 14 different yoga postures

0.2, width shift and height shift in the range of 0.12, brightness at range [1, 1], and horizontal flip. Figure 2 shows an example of each augmentation parameter.

3.3 Transfer learning for yoga posture classification

3.3.1 Feature extraction

Transfer learning (TL) is the learning technique based on previous knowledge, to enable transfer to another domain for classification and feature extraction purposes [27]. TL trains the pre-trained model from a general CNN architecture, with a large number of datasets, to predict with a new small target dataset in the related task. Generally, the CNN model consists of feature extraction and classification layer, in which a feature extraction layer is used to train with large data (ImageNet dataset) and fine-tune the classification layer to predict the target dataset, using a transfer learning technique.

In this study, we used six different pre-trained models, including VGG16, VGG19, MobileNet, MobileNetV2, InceptionV3, and DenseNet201 [28–32], with separately trained weights, using the ImageNet dataset [33], to extract our yoga posture features. The training process begins with the weights of each model-based and freezes the top layers of the pre-trained model to tune the model for our yoga posture dataset. Figure 3 shows the transfer learning technique process flow used in this study.

3.3.2 Deep neural network classifier

After extracting features from the pre-trained model, the classification layer of each model was modified to the new fully connected layer to turn old features into predictions of our 14 categories of yoga posture. Twenty percent of the dropout was applied between the fully connected layers, and the normalized output

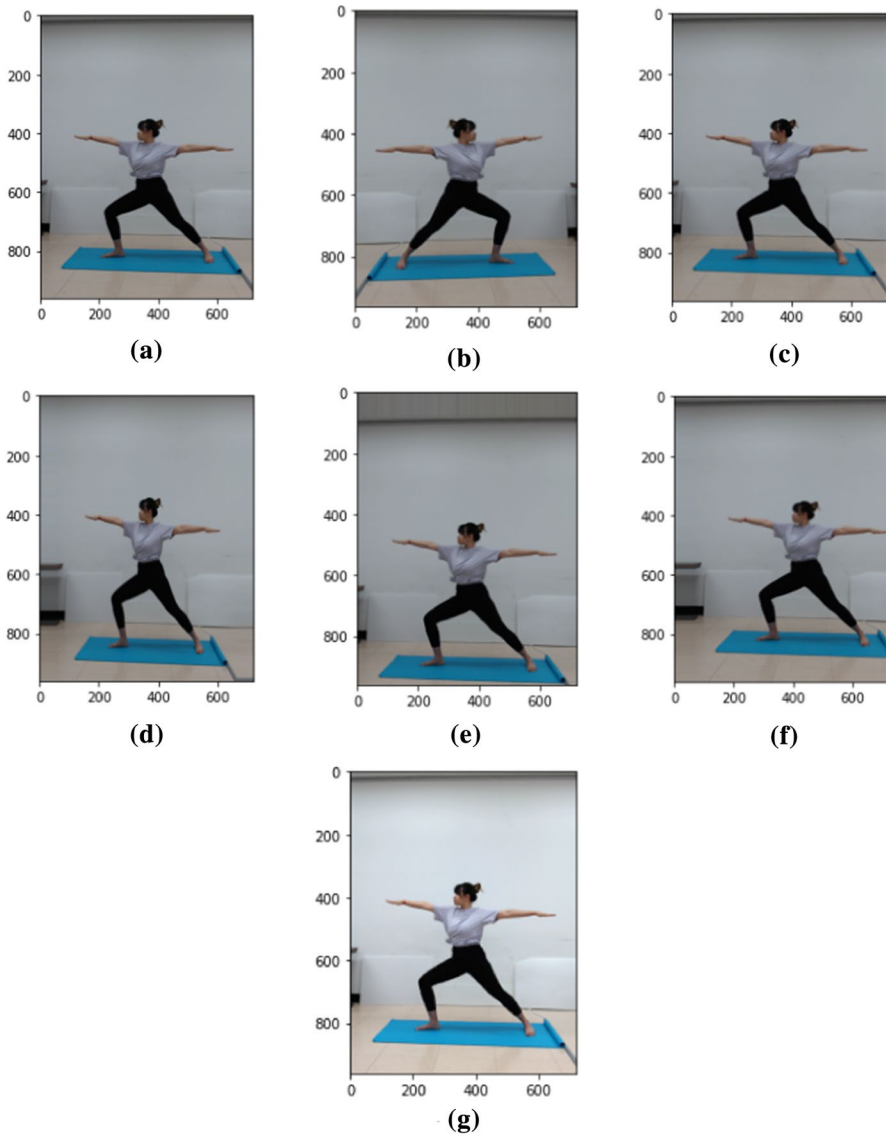


Fig. 2 Examples of data augmentation in this study. **a** Original, **b** Horizontal flip, **c** Shear, **d** Zoom, **e** Height shift, **f** Width shift, and **(g)** Brightness

utilized the softmax function for multi-class classification. Moreover, the network compiled in this study used the categorical cross-entropy loss function and Adam optimizer, with 100 epochs, in the training process. To prevent the overfitting of classification training, the validation loss value is tracked to break the training process, when it is not reduced, by using early stopping with a min delta value of 0.001 at patience 10.

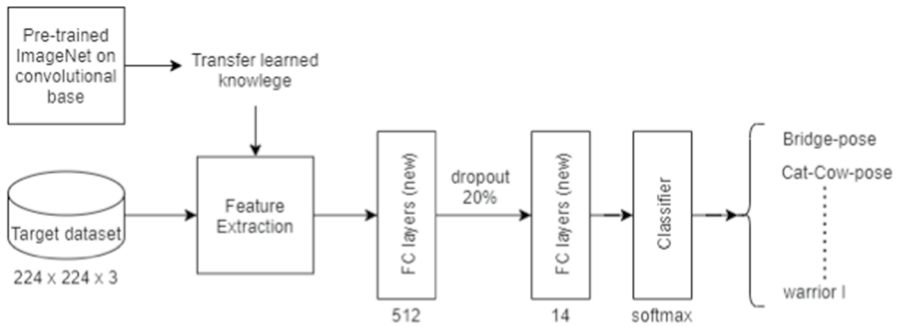


Fig. 3 Transfer learning process from pre-trained on ImageNet dataset to learn new target datasets

3.3.3 Model training setting

In this experimental setup, we utilized the training model in the Python programming language with the main libraries TensorFlow, Keras, OpenCV, Numpy, PIL, and Scikit-Learn. The entire yoga images were resized to a resolution of 224×224, for each of the three channels, to ensure that they were suitable for use in model training. The operating system, adopted for model training of yoga posture classification, featured a dual-core set up: 2× Intel Xeon Silver 4114 CPUs @ 2.20 GHz, 12×16 GB DIMM DDR4 Synchronous RAMs @ 2400 MHz, 3×512 GB Samsung 970 NVMe M.2 SSDs, and 3×NVIDIA TITAN RTX GPUs 24 GB GDDR6 @ 1770 MHz-4608 Compute Unified Device Architecture (CUDA) cores.

3.4 Performance matrix

The performance matrix is used to evaluate the classification, based on four confusion matrix categories: true positive (TP), false positive (FP), true negative (TN), and false negative (FN), by following a multi-class classification table based on [34]. Overall accuracy (OA) presented the overall percentage of classification, where all samples were correctly predicted. The sensitivity identified the ability to classify samples as true positive, and specificity present the classifying as true negative [35]. The Matthews correlation coefficient (MCC) represented a correlation between what was observed and the classification, where an MCC value close to +1 indicates to perfect prediction, −1 reflects disagreement between the prediction and reality and 0 means random classification [36, 37]. To evaluate the yoga posture classification performance, we measured OA, sensitivity, specificity, and MCC, based on the confusion matrix, using Eqs. 1, 2, 3, and 4, respectively, where n is the total number of yoga postures.

$$OA = \frac{TP_1 + TP_2 + \dots + TP_n}{TotalTestingSet} \times 100 \quad (1)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100 \quad (2)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \times 100 \quad (3)$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FN)(FP + TN)(TP + FP)(TN + FN)}} \quad (4)$$

3.5 Yoga self-coaching system

3.5.1 Coaching system feedback

For instructional feedback, the keypoint of the human body was estimated using the Mediapipe algorithm [38]. We extracted body landmarks of each of the keypoints to enable angle calculation of each body joint, e.g., left/right shoulder, left/right elbow, left/right wrist, left/right hip, left/right knee, and left/right ankle. In order to provide instructions for incorrect posture, each joint angle was calculated from three points using Eq. 5.

$$\theta = \frac{180 \times (\text{atan2}((c_y - b_y), (c_x - b_x)) - \text{atan2}((a_y - b_y), (a_x - b_x)))}{\pi} \quad (5)$$

Where c, b, and a is the first, middle, and last point of the three-point, respectively. And it used to identify the angle at this point.

In the yoga coaching system process, yoga posture recognition was predicted based on 14 different yoga postures. When the user performs an incorrect posture following the selected yoga image guide, the joint angle is calculated based on keypoints estimated through utilizing the angle-checking condition. Figure 4 shows the process flow of the real-time yoga coaching system used in this study.

3.5.2 Yoga self-coaching prototype

In this study, we implemented a yoga self-coaching system, connected to an RGB camera. We proposed embedding the optimal yoga posture classification model, investigated in this study, using the Jetson Nano Development Kit for yoga self-exercise at home. The Jetson Nano Development Kit is a microprocessor produced for developing and learning in AI research [39]. We also established a yoga self-coaching mirror based on a Jetson Nano processor connected to a USB camera, called Logitech c920 pro webcam, to recognize the yoga posture in real-time. Moreover, the required distance for using the yoga self-coaching system is 2.5 meters.

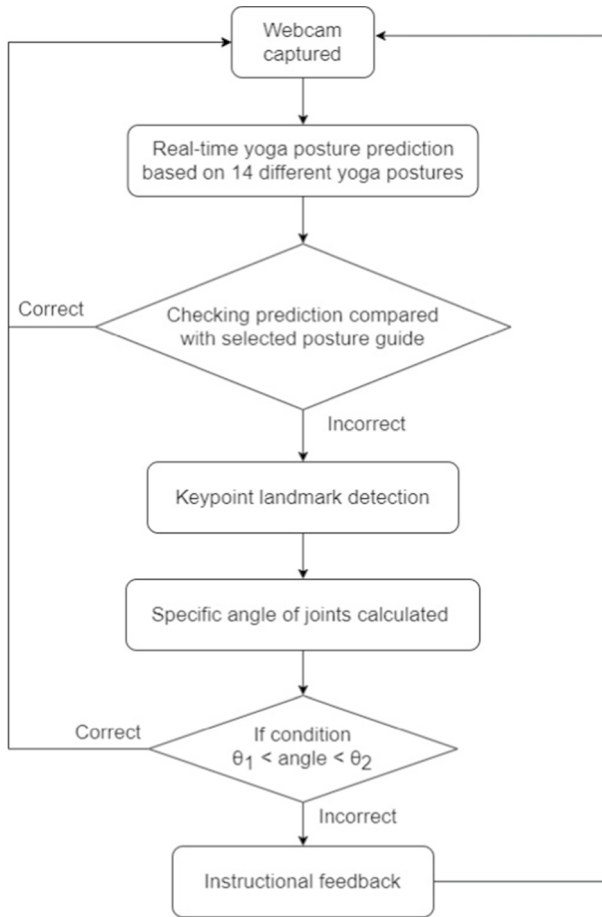


Fig. 4 Real-time coaching feedback flowchart of yoga self-coaching system

4 Results

4.1 Comparison between models in yoga posture classification

After investigating the transfer learning models for yoga posture classification, Table 1 shows the comparison of results between the six transfer learning models used in this study. The transfer learning model based on pre-training of MobileNet with DA achieved an overall accuracy of 98.43%, where the TL-VGG16-DA, TL-VGG19-DA, TL-MobileNetV2-DA, TL-IceptionV3-DA, and TL-DenseNet201-DA performances were 94.90%, 94.90%, 92.16%, 91.76%, and 98.04%, respectively. The sensitivity of TL-MobileNet-DA was the highest (98.30%), while the specificity was 99.88%. The correlation of the TL-MobileNet-DA model classification achieves an MCC value of 0.9831 close to +1, which means that this model achieves accurate

Table 1 Comparison of competitive models in this study

Model	OA ¹	A_Sensitivity ²	A_Specificity ³	MCC ⁴
TL-VGG16-DA	94.90%	94.63%	99.60%	0.9446
TL-VGG19-DA	94.90%	94.19%	99.60%	0.9425
TL-MobileNet-DA	98.43%	98.30%	99.88%	0.9831
TL-MobileNetV2-DA	92.16%	91.35%	99.39%	0.9111
TL-InceptionV3-DA	91.76%	91.07%	99.36%	0.91
TL-DenseNet201-DA	98.04%	97.94%	99.85%	0.9775

¹OA: Overall Accuracy, ²A_Sensitivity: Average Sensitivity, ³A_Specificity: Average Specificity, ⁴MCC: Matthews Correlation Coefficient

predictions of yoga posture classification. However, the TL-InceptionV3-DA model achieved poor classification of our yoga posture dataset, with an overall accuracy of 91.76%, the sensitivity of 91.07%, the specificity of 99.36%, and MCC value of 0.91. According to the classification results from Table 1, transfer learning on MobileNet with DA was recommended for use in the yoga self-coaching system.

Table 2 summarizes the classification performance of sensitivity, specificity, and MCC for each posture from the TL-MobileNet-DA model. Each performance matrix of the TL-MobileNet-DA model was evaluated based on the confusion matrix, as shown in Fig. 5. It provided the highest sensitivity, at 100%, in relation to bridge posture, cat-cow posture, child posture, downward-facing dog posture, sitting posture, extended side angle posture, seated forward bend posture, tree posture, triangle posture, and warrior II posture, while the lowest sensitivity,

Table 2 Classification performance for each posture from the TL-MobileNet-DA model

Posture	Sensitivity	Specificity	MCC ¹	OA ²
Bridge posture	100%	100%	1	
Cat-cow posture	100%	99.57%	0.9738	
Child posture	100%	99.57%	0.9738	
Cobra posture	95%	100%	0.9726	
Corpse posture	93.75%	100%	0.9662	
Downward-facing dog posture	100%	100%	1	
Sitting posture	100%	100%	1	
Extended side angle posture	100%	100%	1	
Plank posture	93.33%	100%	0.9640	
Seated forward bend posture	100%	99.58%	0.9681	
Tree posture	100%	99.56%	0.9776	
Triangle posture	100%	100%	1	
Warrior II posture	100%	100%	1	
Warrior I posture	94.11%	100%	0.9681	98.43%
Average	98.30	99.88	0.9831	–

¹ MCC: Matthews Correlation Coefficient, ² OA: Overall Accuracy

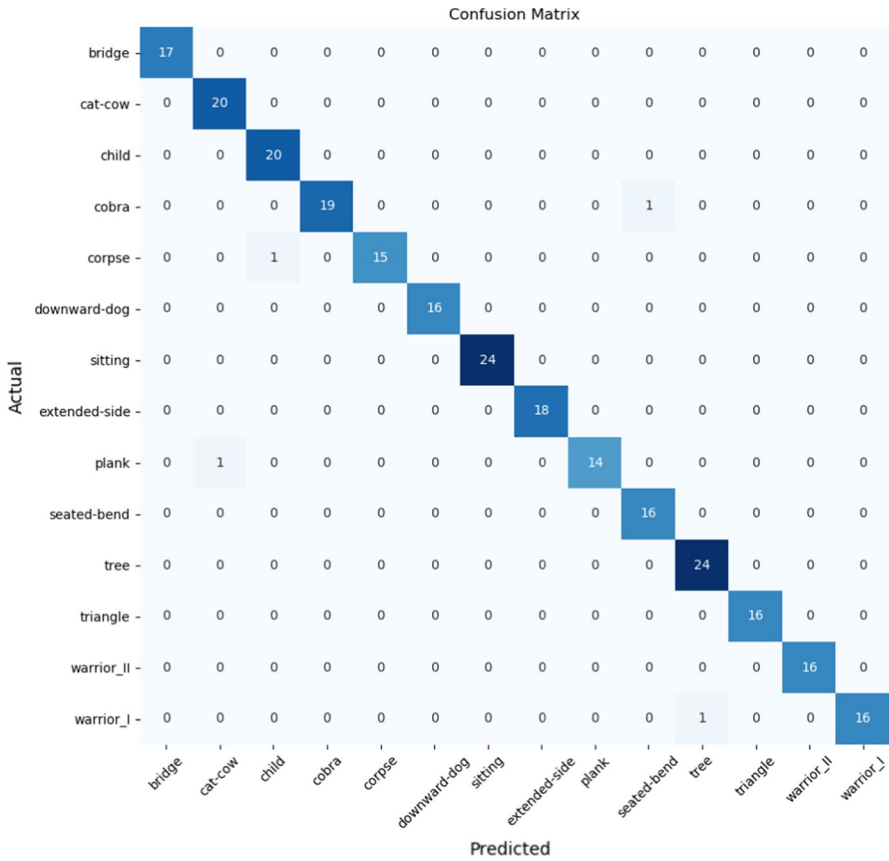


Fig. 5 Confusion matrix of the testing set from the TL-MobileNet-DA model

93.33%, was achieved for plank posture. Moreover, the highest specificity of 100% was achieved for bridge posture, cobra posture, corpse posture, downward-facing dog posture, sitting posture, extended side angle posture, plank posture, triangle posture, warrior II posture, and warrior I posture, while the lowest specificity, 99.56%, was found for tree posture. The perfect prediction of each posture obtains an MCC value of 1 and this was achieved for bridge posture, downward-facing dog posture, sitting posture, extended side angle posture, triangle posture, and warrior II posture. Figure 6 shows the normalized confusion for yoga posture prediction using the TL-MobileNet-DA model. For the normalized confusion matrix performance, perfect predictions were achieved for 10 postures out of the 14 yoga postures included in this study. These included bridge posture, cat-cow posture, child posture, downward-facing dog posture, sitting posture, extended side angle posture, seated forward bend posture, tree posture, triangle posture, and warrior II posture.

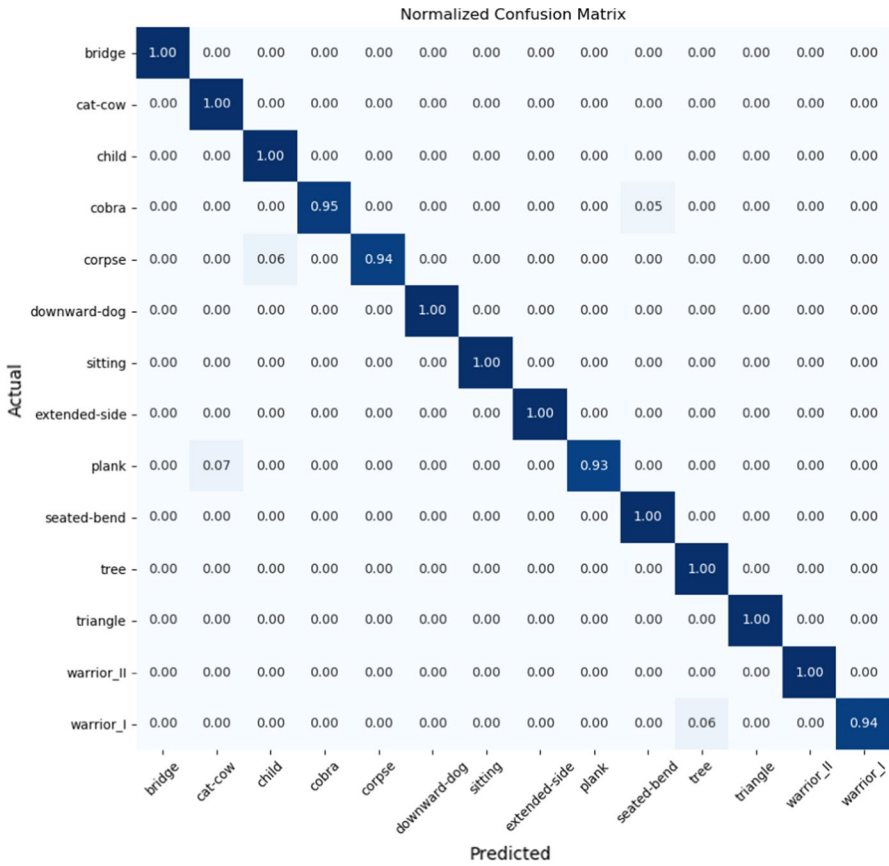


Fig. 6 Normalized confusion matrix from prediction using the TL-MobileNet-DA model

4.2 Development of yoga self-coaching system

The yoga self-coaching system has been completely developed in the Python programming language, with a Python interface based on the Tkinter package. Tkinter is the standard Python interface for the Tk graphical user interface (GUI) toolkit [40]. Figure 7 shows the user interface of the yoga self-coaching system by capturing the user’s posture from a USB camera at the top of the mirror panel and then recognizing the yoga posture, based on the TL-MobileNet-DA model embedded in the Jetson Nano kit. A user can choose the specific yoga posture of 14 different postures studied in this paper to follow the yoga practice. When a user confirms their selected posture for evaluation, the guide yoga image is displayed on the system interface. The user is then able to check for the correct posture by viewing the guidance image, demonstrated in real-time, to achieve posture recognition. The resulting recognized posture of the user will pop up with a success message of “Correct :)” when the user achieves in the right posture and



Fig. 7 Snapshots of the yoga self-coaching system

illustrated the fail message of “Incorrect :(” when the user has not achieved the correct posture. The recognized yoga label from our system is shown below the performance checking results, as shown in Fig. 7. Furthermore, the system will provide instructional feedback to a user, to provide guidance enabling them to perform the posture correctly, based on the angle of each joint, calculated from keypoints extracted when a user performs the incorrect yoga posture.

5 Discussion

In the literature review, several studies have proposed an algorithm and applied many techniques to recognize yoga postures based on ML or DL [19–25]. According to the small amount of yoga posture data available on the Internet, other researchers collected their yoga posture data. Therefore, in our experiment, we collected the yoga posture dataset in our indoor laboratory with an RGB camera at 14 different yoga postures. In order to recognize the yoga posture, we used a different approach from previous studies and applied the transfer learning with

a general CNN architecture to pre-train for feature extraction and modified the classifier layer to predict our yoga posture dataset.

The results indicated that transfer learning on MobileNet worked well with DA on our yoga posture dataset by enabling a better overall accuracy of prediction than achieved by other competitive models in this study. The classification evaluated is the truthful score with the coefficient value of the MCC value, which is close to +1. Although the experiment of the TL-MobileNet-DA model showed a slight inaccuracy in predicting the plank posture, the recognition model is still acceptable for embedding within a yoga self-coaching system.

Due to inaccessibility to the same yoga posture dataset, direct comparison with the literature is impossible. The work in [23] provided a self-training system for posture recognition of three postures: tree, warrior III, and downward-facing dog, while our consideration of yoga postures was 14 asanas and these were collected with a normal RGB webcam. The five participants in the data collection performed each of the three yoga postures 5 times, while in our case, eight participants volunteered to perform all 14 yoga postures 10 times. The overall accuracy evaluated by their method was 82.84%. Moreover, Chen et al. [24] used at the skeleton with body contour to extract the feature points of twelve yoga postures, collected from the Kinect sensor, and achieved an accuracy of 93.45%. However, prior studies used manual feature extraction and made a separate model for each yoga posture, which is time-consuming to implement when appending new postures. Incorporating new classes of yoga postures when adopting our approach is possible through adding more neurons in the last dense layer and re-training on the new dataset. Our method achieved an accuracy of 98.43%, which is superior to that of Chen et al. On the other hand, our work comprises a coaching system, called yoga self-coaching system, with real-time instructional feedback, based on the calculated angle of body joints using Mediapipe keypoints extraction.

Reviewing the products inspired by several companies related to yoga assistance, the Yoga Pants, SmartMat, and YogaNotch [14–16] do not enable the viewing of the users' reflection while doing the exercises; the MIRROR [17] does enable the viewing of one's reflection but does not provide any posture recognition. Our proposed a yoga self-coaching system plays a significant role in solving these problems.

Nonetheless, our study established a yoga system that recognized 14 different yoga postures for beginner yoga practice. The different postures provide different health benefits. Thus, more yoga postures need to be investigated in the future to improve our proposed system.

6 Conclusions

This paper presented a yoga self-coaching system based on transfer learning techniques. The first step of this study was to collect the yoga posture dataset using a normal RGB webcam and then to apply data augmentation techniques. The transfer learning technique, pre-trained on the MobileNet model, was investigated. In the last phase, we established an AI yoga system utilizing a prediction model for real-time inference.

In conclusion, the yoga posture classification method achieved a performance accuracy of 98.43%, which was accomplished by embedding this within our yoga

self-coaching system. The yoga self-coaching system was developed to recognize the yoga postures following the selected yoga posture guide and to output the predicted result and give real-time guidance for incorrect postures. The identification of incorrect posture is based on the calculated angle of the joints achieved through keypoint estimation using the Mediapipe algorithm.

In summary, we developed a yoga self-coaching system that can predict yoga posture and confirm instruction feedback in real time. Since the start of Covid-19, home training has increased and our developed system supports this in our opinion. The yoga self-coaching system is used to recognize the correct yoga posture and provide instruction in real time.

Acknowledgements This research was supported by a grant of the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (grant number: HI21C1831) and the Soonchunhyang University Research Fund.

References

1. Basavaraddi IV (2015) Yoga: it origin, history and development. Government of India, Ministry of External Affairs, p 23
2. Balasubramaniam M, Telles S, Doraiswamy PM (2013) Yoga on our minds: a systematic review of yoga for neuropsychiatric disorders. *Front Psychiatry* 3:117. <https://doi.org/10.3389/fpsy.2012.00117>
3. Sherman KJ, Cherkin DC, Erro J, Miglioretti DL, Deyo RA (2005) Comparing yoga, exercise, and a self-care book for chronic low back pain: a randomized, controlled trial. *Ann Intern Med* 143:849–856. <https://doi.org/10.7326/0003-4819-143-12-200512200-00003>
4. Patel SR, Zayas J, Medina-Inojosa JR, Loprinzi C, Cathcart-Rake EJ, Bhagra A, Olson JE, Couch FJ, Ruddy KJ (2021) Real-world experiences with yoga on cancer-related symptoms in women with breast cancer. *Glob Adv Health Med* 10:2164956120984140. <https://doi.org/10.1177/2164956120984140>
5. Fishman L, Saltonstall E, Genis S (2009) Understanding and preventing yoga injuries. *Intern J Yoga Therapy* 19:47–53
6. Swain TA, McGwin G (2016) Yoga-related injuries in the United States from 2001 to 2014. *Orthop J Sports Med* 4:2325967116671703. <https://doi.org/10.1177/2325967116671703>
7. Russell K, Gushue S, Richmond S, McFaul S (2016) Epidemiology of yoga-related injuries in Canada from 1991 to 2010: a case series study. *Int J Inj Contr Saf Promot* 23:284–290. <https://doi.org/10.1080/17457300.2015.1032981>
8. Bilal M, Maqsood M, Yasmin S, Hasan NU, Rho S (2021) A transfer learning-based efficient spatiotemporal human action recognition framework for long and overlapping action classes. *The J Supercomput.* <https://doi.org/10.1007/s11227-021-03957-4>
9. Bukhari M, Bajwa KB, Gillani S, Maqsood M, Durrani MY, Mehmood I, Ugail H, Rho S (2020) An efficient gait recognition method for known and unknown covariate conditions. *IEEE Access* 9:6465–6477. <https://doi.org/10.1109/ACCESS.2020.3047266>
10. Rusdorf S, Brunnett G, Lorenz M, Winkler T (2006) Real-time interaction with a humanoid avatar in an immersive table tennis simulation. *IEEE Trans Vis Comput Graph* 13:15–25. <https://doi.org/10.1109/TVCG.2007.18>
11. Reddy MS, Venkatramana SA, Ramji BR (2015) e-Yoga prescription designed for computer users using e-Yoga environment for posture. In: *Proceedings of the 3rd International Congress on Sport Sciences Research and Technology Support*, pp. 48–52
12. Chiddarwar GG, Ranjane A, Chindhe M, Deodhar R, Gangamwar P (2020) AI-based yoga pose estimation for android application. *Int J Inn Scien Res Tech* 5:1070–1073
13. Anusha M, Dubey S, Raju PS, Pasha IA (2019) Real-time yoga activity with assistance of embedded based smart yoga mat. In: *2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication*. <https://doi.org/10.1109/IESPC.2019.8902371>
14. Wearable X. <https://www.wearablex.com/>. Accessed 16 July 2021
15. SmartMat. <https://www.smartmat.com/about/>. Accessed 16 July 2021
16. YOGANOTCH. <https://yoganotch.com/>. Accessed 16 July 2021

17. MIRROR. <https://www.mirror.com/>. Accessed 16 July 2021
18. Cao Z, Hidalgo G, Simon T, Wei SE, Sheikh Y (2019) OpenPose: realtime multi-person 2D pose estimation using part affinity fields. *IEEE Trans Pattern Anal Mach Intell* 43:172–186. <https://doi.org/10.1109/TPAMI.2019.2929257>
19. Zhang Z (2012) Microsoft kinect sensor and its effect. *IEEE Multi* 19:4–10. <https://doi.org/10.1109/MMUL.2012.24>
20. Karbasi M, Bhatti Z, Nooralishahi P, Shah A, Mazloomnezhad SM (2015) Real-time hands detection in depth image by using distance with Kinect camera. *Int J Inter Things* 4:1–6. <https://doi.org/10.5923/c.ijit.201501.01>
21. Manap MS, Sahak R, Zabidi A, Yassin I, Tahir NM (2015) Object detection using depth information from Kinect sensor. In: 2015 IEEE 11th International Colloquium on Signal Processing & Its Applications. pp 160–163. <https://doi.org/10.1109/CSPA.2015.7225638>
22. Charles J, Pfister T, Magee D, Hogg D, Zisserman A (2016) Personalizing human video pose estimation. *arXiv arXiv:1511.06676*
23. Chun J, Park S, Ji M (2018) 3D human pose estimation from RGB-D images using deep learning method. In: Proceedings of the 2018 International Conference on Sensor, Signal and Image Processing, pp 51–55. <https://doi.org/10.1145/3290589.3290591>
24. Chen HT, He YZ, Hsu CC (2018) Computer-assisted yoga training system. *Multi Tools App* 77:23969–23991. <https://doi.org/10.1007/s11042-018-5721-2>
25. Verma M, Kumawat S, Nakashima Y, Raman S (2020) Yoga-82: a new dataset for fine-grained classification of human poses. *arXiv arXiv:2004.10362*
26. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. *J Big Data* 6:1–48. <https://doi.org/10.1186/s40537-019-0197-0>
27. Orenstein EC, Beijbom O (2017) Transfer learning and deep feature extraction for planktonic image data sets. In: 2017 IEEE Winter Conference on Applications of Computer Vision, pp 1082–1088. <https://doi.org/10.1109/WACV.2017.125>
28. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition *arXiv arXiv:1409.1556*
29. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Efficient convolutional neural networks for mobile vision applications. *arXiv arXiv:1704.04861*
30. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2019) Mobilenetv2: Inverted residuals and linear bottlenecks. *arXiv arXiv:1801.04381*
31. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2015) Rethinking the inception architecture for computer vision. *arXiv arXiv:1512.00567*
32. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2018) Densely connected convolutional networks. *arXiv arXiv:1608.06993*
33. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Adv Neu Inf Process Sys* 25:1097–1105. <https://doi.org/10.1145/3065386>
34. Kutlu Y, Kuntalp D (2009) Feature reduction method using self organizing maps. In: 2009 International Conference on Electrical and Electronics Engineering-ELECO 2009, pp II–129. <https://doi.org/10.1109/ELECO.2009.5355238>
35. Trevethan R (2017) Sensitivity, specificity, and predictive values: foundations, liabilities, and pitfalls in research and practice. *Front Public Health* 5:307. <https://doi.org/10.3389/fpubh.2017.00307>
36. Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 21:1–13. <https://doi.org/10.1186/s12864-019-6413-7>
37. Boughorbel S, Jarray F, El-Anbari M (2017) Optimal classifier for imbalanced data using Matthews correlation coefficient metric. *PLoS One* 12:e0177678. <https://doi.org/10.1371/journal.pone.0177678>
38. Bazarevsky V, Grishchenko I, Raveendran K, Zhu T, Zhang F, Grundmann M (2020) BlazePose: On-device real-time body pose tracking. *arXiv arXiv:2006.10204*
39. Jetson Nano Developer Kit. <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>. Accessed 16 July 2021
40. Tkinter – Python interface to Tcl/Tk. <https://docs.python.org/3/library/tkinter.html>. Accessed 16 July 2021